Movement recognition and prediction using DMPs

Ali H. Kordia[1] Francisco S. Melo[2]

***Abstract*— This paper proposes an approach for (a) recognizing an observed trajectory from a library of pre-learned motions; and (b) predicting the target position of such trajectory. In our approach, motions are represented as Dynamic Movement Primitives (DMPs). We use critical points from the observed trajectory to time-align it with those in the library. To match the observed trajectory with those in the library, we compare the changes in velocity orientation between consecutive critical points. The proposed approach is computationally light and, as such, can be performed at execution time. As for the prediction, we adopt a similar approach: after matching the observed trajectory to one in the library, we use the latter to predict the target point, modulating it to match the observed trajectory. Both recognition and prediction approaches are probabilistic and, as such, provide a measure of certainty/uncertainty in the recognition/prediction process. Such a measure of uncertainty is important in tasks involving human-robot collaboration, as it allows the robot to decide when it is sufficiently certain to act conditioned on the estimated trajectory. We illustrate our approach both in simulation and in a human-robot interaction scenario involving the Baxter robot.**

I. INTRODUCTION

One of the most important aspects of human-robot collaboration is the ability of the robot to recognize what the human partner is doing, and move accordingly. In human-human collaboration, each person observes and "intuitively" recognizes the partner's motion, which in turn allows the person to act in the best possible way. Humans are naturally flexible with respect to actions of a partner and able to adapt, fostering high-level collaboration and inter-partner trust.

In order for such high-level collaboration to be possible, robots must also be able to recognize and adapt to the observed movement of a human partner. In this paper, we propose an approach for trajectory recognition and prediction. We represent motions as *dynamic movement primitives* (DMP) [1] and propose a novel method that, given a library of previously observed movements, identifies a partially observed trajectory from among those in the library, predicting its (unobserved) target point. The fact that our approach does not require the full trajectory to be observed is fundamental, as it can be used in run-time for actual human-robot interaction scenarios. Our approach can be broken down into its two fundamental steps: understanding/recognizing an observed trajectory, and inferring its target.

There are several works in the literature that address the problem of trajectory recognition in the context of human-robot interaction, and many of these works use various forms of Hidden Markov Model (HMM) [2], [3], [4]. However, such methods require big data sets for training, which are costly to obtain in human-robot collaboration scenarios. As an example, Yoon et al [5] propose an approach to recognize gestures from video data. They describe a gesture with a number of features that are invariant to translation, rotation and scaling (e.g., location, velocity, angle of motion) and use these data to build a Hidden Markov Model to recognize various kinds of hand gestures. In another work, Black and Jepson [6] use a condensation algorithm to compute a probability distribution over a set of models given the input trajectory. The distribution is computed from the difference between the amplitude of the input and model motions, after adequate time scaling. The computed distribution can then be used to combine the prediction of the different models. The same approach was also tested in a face recognition scenario to recognize facial expressions from facial motions [7].

Also related is the work of Kherallah et al [8], where stroke movement is used to recognize Arabic digits. In particular, the authors propose the use of Beta-elliptical representations to describe the writing motion, and then use a neuro-fuzzy system for digit recognition. Maeda et al [9] introduce *interactive probabilistic movement primitives* (I-ProMPs) to represent collaborative human-robot motions. In this work, the authors assume that all motions have a constant rate of change and, therefore, time alignment can be achieved by a simple scaling. This reduces recognition to the estimation of human action phase, circumventing the need for time alignment.

Dermy et al [10] use a set of pre-learned motor primitives (MP) for recognition. The use of an MP library is similar to our own approach. However, unlike our work, the authors assume that the observed movement is an exact match to one of the pre-learned MPs, thus disregarding some key challenges in motion recognition (such as time alignment).

The I-ProMPs of Maeda et al [11] represent collaborative trajectories between a human and a robot by using a probabilistic model, capturing through standard regression a Gaussian representation for both the human and robot trajectories. At recognition time, the observed trajectory—corresponding to the human action—is used to compute the corresponding robot trajectory by a simple probabilistic operation (conditioning). However, in order to build a robust I-ProMP model, a rich set of demonstrations is required. Their approach is also sensitive to speed variability. Our proposed approach is more robust to speed changes and, moreover, does not require the observation of the full human trajectory to compute the robot's movement.

[1]Ali Hani Kordia is with INESC-ID and with Instituto Superior Técnico, University of Lisbon, Portugal; ali.kordia@tecnico.ulisboa.pt.

[2]Francisco S. Melo is with INESC-ID and with Instituto Superior Técnico, University of Lisbon, Portugal; fmelo@inesc-id.pt.

Perhaps closest to our own approach is that of Pérez-D'Arpino and Shah [12]. In their approach they also use a library of learned movements and use time series analysis to estimate the target of the observed motion. The estimation requires time alignment of the observed trajectory to those in the library, which is achieved through online dynamic time warping (DTW). Since this process is time-consuming, they use a parallel architecture to perform the DTW with the different motions in the library. However, unlike the approach proposed herein, they do not support target modulation, assuming that the observed motions are simply perturbed versions of those in the library. Along the same lines, Mainprice and Berenson [13] also perform recognition from a library of known movements. They use *occupancy* for movement recognition, overcoming the need for time alignment but, once again, assuming that the observed motions are perturbed versions of those in the library.

Finally, other works have explored motion prediction for human-robot interaction. Mainprice et al. [14] consider prediction in the context of a single task, to avoid interference between the robot's motion and that of the human. Prada et al. [15] consider handover tasks, where the movement of the robot is adjusted in runtime to that of the human.

Our proposed approach comprises two steps: in a first step, we rely on the notion of *critical points* [16] for time alignment and recognition. In particular, we compare the observed trajectory with those in the library at only those critical points, thus avoiding more computationally intensive time alignment procedures (such as DTW) or too stringent assumptions (such as linear time scaling between the observed trajectory and those in the library). We compute the probability of the observed trajectory given each movement in the library assuming a maximum entropy model, and use the probabilistic representation to assess the level of certainty of the system on the recognition. Target prediction is then the expected target of the different motions in the library.

II. Recognition and Prediction Using Critical Points

We address the problem of observing a (partial) trajectory and recognizing it as being part of one of a set of movements in a library, each represented as a DMP. Once the DMP that best matches the observed trajectory is identified, we want to predict the end-point of the trajectory. We consider each of the two problems (recognition and prediction) separately.

A. Recognition

Ideally, our approach to recognition should be able to accommodate trajectories that differ from those in the library in the (linear and angular) velocity of execution, and changes in the time scale and target. We therefore adopt a representation that is invariant to those transformations.

Given a trajectory $\mathbf{y}_{0:T} = \{y(t), t = 0, \dots, T\}$, *critical points* are defined as minima of the function

$$h(t) = \|\dot{y}(t)\|^2 + \|\ddot{y}(t)C_t\|^2,$$

where C_t is a time-dependent weighting factor.[1] The minima of h correspond to "inflection" points of the trajectory. Such critical points have been proposed as suitable candidates for trajectory segmentation [16].

In this work, we describe a segment between two critical points by its change of velocity and of orientation. Given a trajectory $\mathbf{y}$,[2] with critical points $\tau_0, \dots, \tau_{M+1}$, we compute

- The velocity, $v(\tau_m)$, at each of the critical points;
- The orientation, $\theta(\tau_m)$, at each of the critical points.

Segment m, between critical points τ_m and τ_{m+1}, is then described by the tuple

$$s^{(m)}(\mathbf{y}) = \big(\delta_v^{(m)}(\mathbf{y}), \delta_\theta^{(m)}(\mathbf{y})\big), \qquad m = 0, \dots, M,$$

where

$$\delta_v^{(m)}(\mathbf{y}) = v(\tau_{m+1}) - v(\tau_m)$$

and

$$\delta_\theta^{(m)}(\mathbf{y}) = \theta(\tau_{m+1}) - \theta(\tau_m).$$

By considering a trajectory only at critical points, we overcome the need for explicit time alignment. Additionally, to render the representation invariant to rotations, scaling and target, we define the normalized segment representation

$$\bar{s}^{(m)}(\mathbf{y}) = \big(\bar{\delta}_v^{(m)}(\mathbf{y}), \bar{\delta}_\theta^{(m)}(\mathbf{y})\big), \qquad m = 1, \dots, M.$$

with

$$\bar{\delta}_v^{(m)}(\mathbf{y}) = \frac{\delta_v^{(m)}(\mathbf{y})}{\delta_v^{(0)}(\mathbf{y})}, \qquad \bar{\delta}_\theta^{(m)}(\mathbf{y}) = \frac{\delta_\theta^{(m)}(\mathbf{y})}{\delta_\theta^{(0)}(\mathbf{y})}.$$

A trajectory $\mathbf{y} = \{y(t), t = 0, \dots, T\}$, with critical points $\tau_0, \dots, \tau_{M+1}$ is thus summarized as a sequence

$$S(\mathbf{y}) = \{\bar{s}^{(m)}(\mathbf{y}), m = 1, \dots, M\}.$$

Now for a library of N trajectories represented as DMPs,[3] $\{\mathbf{y}_n, n = 1, \dots, N\}$, let

$$S(\mathbf{y}_n) = \{\bar{s}^{(m)}(\mathbf{y}_n), m = 1, \dots, M_n\}$$

denote the summarized representation of $\mathbf{y}_n$. Given an arbitrary (partial) trajectory $\mathbf{y}_{0:t}$, described by the summarized representation

$$S(\mathbf{y}_{0:t}) = \{\bar{s}^{(m)}(\mathbf{y}_{0:t}), m = 1, \dots, M_*\},$$

we compute the error between segment m of $\mathbf{y}_{0:t}$ and $\mathbf{y}_n$ as

$$\epsilon_n^{(m)} = \left\|\bar{\delta}_v^{(m)}(\mathbf{y}_n) - \bar{\delta}_v^{(m)}(\mathbf{y}_{0:t})\right\| + \left\|\bar{\delta}_\theta^{(m)}(\mathbf{y}_n) - \bar{\delta}_\theta^{(m)}(\mathbf{y}_{0:t})\right\|.$$

Finally, we define the recognition error of DMP $\mathbf{y}_n$ to be

$$E_n = \sum_{m=1}^{M_*} \epsilon_n^{(m)}.$$

We note that, if $M_n < M_*$, then we define $\epsilon_n^{(m)}, m > M_n$, to take a constant large value.

[1] In all our results, we use a weighting factor $C_t = 1$.

[2] To simplify the notation, we omit the time subscripts $_{0:t}$ when immaterial for the discussion.

[3] For ease of presentation, we abusively denote DMPs by their nominal trajectories.

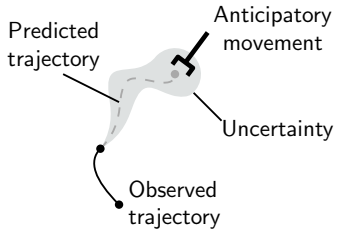


Fig. 1: Prediction and anticipatory movement. The robot observes part of a trajectory and predicts the full trajectory and its target. Once the uncertainty goes below some threshold, the robot can move anticipatorily to the predicted target.

Finally, we use a maximum entropy distribution to assign to each DMP $\mathbf{y}_n$ a value

$$\omega_n = \Pr(\mathbf{y}_n \mid \mathbf{y}_{0:t}) = \frac{\exp(-E_n)}{\sum_{\ell=1}^{N} \exp(-E_\ell)},$$

indicating the probability that it was DMP $\mathbf{y}_n$ to yield the partial trajectory $\mathbf{y}_{0:t}$.

B. Prediction

The purpose of prediction is to infer the target position of an observed movement. Prediction is useful, for example, in the context of human-robot interaction, as it allows the robot to adjust its movement to that of the human user by predicting the target and timing of the later (see Fig. 1). Anticipatory behavior fosters successful collaboration between robot and human, closer to that observed between humans.

We discuss two distinct approaches to prediction. The first is based on the critical point information already used in the recognition (Section II-B.1), while the second relies on the formalism of dynamic motion primitives (DMP) to estimate the target of the motion (Section II-B.2).

$\diamond$

Prediction occurs at run-time, as we make successive observations of the trajectory, and uses the output of the recognition approach described in Section II-A. Let $\mathbf{y}_{0:t}$ denote a partial trajectory, observed up to time step t, and let $w_n(t)$ denote the probability

$$\omega_n(t) = \Pr(\mathbf{y}_n \mid \mathbf{y}_{0:t}),$$

computed as described in Section II-A. In other words, as we observe larger portions of the trajectory $\mathbf{y}_{0:t_1}, \mathbf{y}_{0:t_2}, \ldots$, we compute a sequence of probabilities $\{\boldsymbol{\omega}(t_1), \boldsymbol{\omega}(t_2), \ldots\}$, where each $\omega_n(t)$ is computed from $\mathbf{y}_{0:t}$ and we write $\boldsymbol{\omega}(t)$ to denote the vector with nth entry given by $\omega_n(t)$.

The predicted trajectory at each time step t, $\hat{\mathbf{y}}_{0:T}$, can now be computed as a weighted combination of the trajectories predicted from those in the library as

$$\hat{\mathbf{y}}_{0:T} = \sum_{n=1}^{N} \omega_n(t) \hat{\mathbf{y}}_n, \tag{1}$$

where $\hat{\mathbf{y}}_n$ is the prediction from the nth trajectory in the library. The predicted target of the trajectory corresponds to $\hat{y}(T)$. The uncertainty in the prediction can also be computed as the (co)variance of the prediction at each $\hat{y}(t)$ of $\hat{\mathbf{y}}_{0:T}$,

$$\boldsymbol{\Sigma}(t) = \sum_{n=1}^{N} \omega_n(t) (\hat{y}(t) - \hat{y}_n(t))^\top (\hat{y}(t) - \hat{y}_n(t)).$$

It is worth noting that the longer we observe the trajectory $\mathbf{y}$, the more reliably our approach will be able to estimate the probabilities $\omega_n, n = 1, \ldots, N$, and the more accurate the prediction in (1) will be.

In the continuation, we discuss two ways of computing the predictions $\hat{\mathbf{y}}_n$ according to the movements in the library.

1) Prediction using position ratios: In this method, we again use critical point information to compute the predicted trajectory. In particular, given the partial trajectory $\mathbf{y}_{0:t}$, let τ_m denote the last observed critical point. Let $\mathbf{y}_n$ denote a movement in the library, and define

$$\bar{\delta}_y(\mathbf{y}_n) = \frac{y_n(\tau_{M+1}) - y_n(\tau_m)}{y_n(\tau_m) - y_n(\tau_0)}. \tag{2}$$

We rely on the simple intuition that, if the movement $\mathbf{y}_{0:t}$ that we have observed is, indeed, a partial execution of $\mathbf{y}_n$, then the ratio in (2) remains constant and we have that

$$\frac{y(\tau_{M+1}) - y(\tau_m)}{y(\tau_m) - y(\tau_0)} = \bar{\delta}_y(\mathbf{y}_n).$$

Therefore, we can compute the prediction

$$\begin{aligned} &\hat{y}_n(\tau_{M+1}) \\ &= y(\tau_m) + \frac{y(\tau_m) - y(\tau_0)}{y_n(\tau_m) - y_n(\tau_0)} \cdot (y_n(\tau_{M+1}) - y_n(\tau_m)) \end{aligned}$$

Such predictions are then combined using (1).

2) Prediction target using the DMP equations: In this method, we rely on the representation of trajectories as dynamic motion primitives (DMPs) and use the DMP parameter information (from both the observed trajectory and the DMPs in the library) to build our prediction. From the observed trajectory we can compute the parameters of DMP by using information regarding the starting point, target point (which is unknown), the velocity or the acceleration. The parameters from the DMP are the weights, the canonical system, the basis functions and the constants α and β [1].

Let $\mathbf{y}_n$ denote a DMP in the library. We must select one point from the observed trajectory and use the DMP equation in this state. We select the last observed critical point, τ_m, to estimate the canonical system variable (x) by determining the relative time that has passed from the beginning of the trajectory, yielding

$$x = \exp\left(-\alpha_x \frac{m}{M} T\right),$$

where α_x is a constant of the canonical system, M is the number of critical points, T is a time-scaling factor, m is the selected critical point [17].

We now compute the weights of the basis functions for the DMP $\mathbf{y}_n$, $w_{i,n}$, and the value of those basis functions at

the computed canonical state, $\psi_i(x)$. The dynamic equations of the DMP at the selected critical point become [1]

$$\ddot{y}_n = \alpha_y(\beta_y(g - y_n) - \dot{y}_n) + \frac{\sum_i \psi_i(x) w_{i,n}}{\sum_i w_{i,n}} x(g - y_n(\tau_0)),$$

where α_y, β_y are constants and g is the target. Letting

$$A = \frac{\sum_i \psi_i(x) w_{i,n}}{\sum_i w_{i,n}},$$

we can isolate the target (g) to get our prediction

$$g = \hat{y}_n(\tau_{M+1}) = \frac{\ddot{y} + \alpha_y \dot{y} + \alpha_y \beta_y y + Axy(\tau_0)}{Ax + \alpha_y \beta_y} \quad (3)$$

Applying the approach above to all the DMPs in the library, we get a set of predictions $\hat{y}_n(\tau_{M+1}), n = 1, \ldots, N$, which again can be combined using (1).

C. The Threshold

In real life, humans do not need to wait for a partner to finish its movement to take the decision and start their own action, or watch all of the companion's trajectory to start the next required task. Likewise, we should have our robot act when there is sufficient data to know with some confidence what the movement of the user will be. In our approach, we allow for some margin of error and define a threshold (ϵ) such that, if the level of uncertainty in the observed trajectory's target is below the threshold, the robot will move.

The threshold driving such decision translates the notion of "acceptable error", in terms of the robot's ability to recognize the movement/DMP generating the observed trajectory.

To verify the level of (un)certainty in the recognition we compute the entropy $H(t)$ of the distribution $\{\omega_1(t), \ldots, \omega_N(t)\}$ after each critical point,

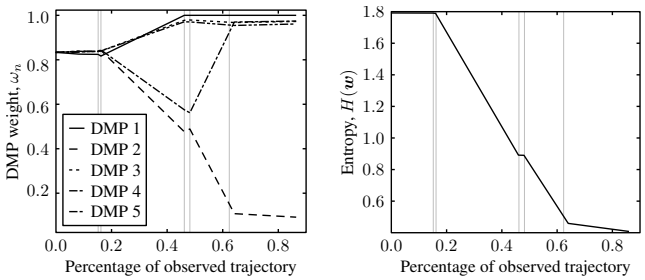
$$H(t) = -\sum_n \omega_n(t) \log \omega_n(t).$$

When the entropy falls below some pre-defined threshold ϵ, the robot moves towards the predicted target.

III. The Experiments

In this section, we test our proposed approach both in simulation and with a real robot. We present three sets of experiments, aimed at answering the following questions: (i) is the proposed approach able to successfully recognize and predict the target point of a trajectory from partial observations? (ii) is the proposed approach successfully able to handle a library with a significant number of motor primitives, providing robust recognition and prediction? (iii) can we successfully use the proposed approach in a real world scenario using the Baxter robot?

To answer the first question, we use a simulation environment where the robot is provided a library with a small number of 2D trajectories. We then show the robot successively larger portions of a trajectory and evaluate the robot's ability to identify the observed trajectory with those in its library, as well as its ability to predict the endpoint of the trajectory. To answer the second question, we run a similar experiment, but where the robot is now endowed with a library of 85 distinct motor primitives. We then show the robot a trajectory that consists of a perturbed version of a randomly selected DMP from the library and observe the robot's ability to identify the movement being performed and predicting the corresponding endpoint. Finally, to answer the third question, we implement our system in a Baxter robot and assess the performance in a real-world task.



(a) Recognition weights ω_n for all DMPs at successive critical points (vertical gray lines).

(b) Entropy of the prediction with the observation of new critical points.

Fig. 2: Weight and entropy evolution during the recognition process, as larger portions of the trajectory are observed.

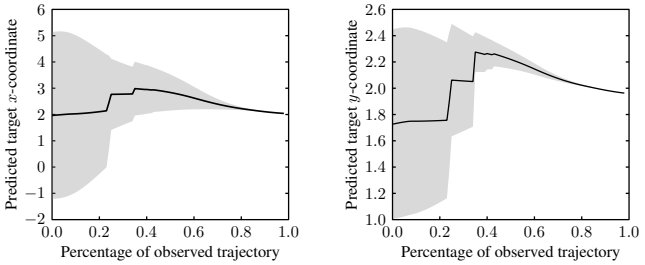
A. Recognition

We start by discussing the results of our first experiment. Figure 2a illustrates the recognition process for a library of $N = 5$ DMPs in two dimensions, as we observe a larger percentage of the trajectory. The plot shows the evolution of the weights, $\omega_n, n = 1, \ldots, 5$, as we observe increasingly larger number of critical points. When a new critical point is observed, the plot exhibits abrupt—confirming that critical points are most informative to identify the movement producing the observed trajectory. In the example, the library includes 4 movements similar to the one being observed and one very different movement. The weights associated with similar movements approach 1, while the weight associated with the different movement steadily decreases.

B. Prediction

We now assess the robot's ability to predict the endpoint of the observed trajectory, following the approaches discussed in Section II-B. Both approaches exhibit similar performance, so we depict only the results obtained with that in Section II-B.1. Figure 3 showcases how the uncertainty evolves as a larger portion of the trajectory is observed. We depict the uncertainty in each of the two coordinates (x and y): The solid line corresponds to the evolution of the predicted target, $\hat{y}(\tau_{M+1})$; the shaded area depicts the variance in the prediction.

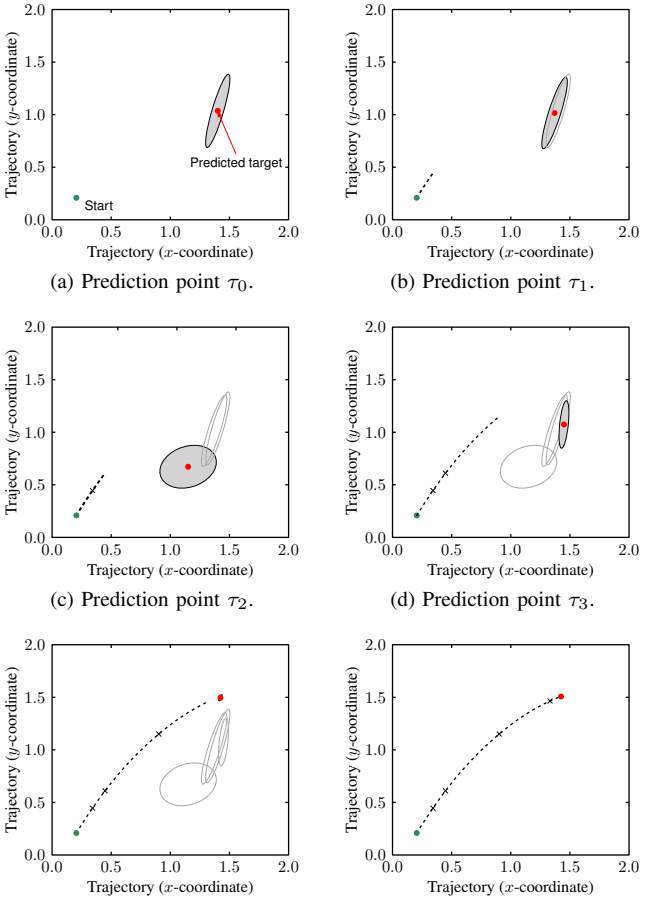
We can observe the improvement in prediction: the variance of the predicted targets greatly decreases as additional critical points are observed. From this figure, we can see that there are sharp changes at the critical points, since the available information for recognition changes significantly at these points. This observation supports our claim that critical points indeed provide useful information for recognition.



(a) Prediction of the target x coordinate.

(b) Prediction of the target y coordinate.

Fig. 3: Uncertainty in the prediction of the target for the two dimensions of the trajectory. The solid line corresponds to the mean predicted target, $\hat{y}(\tau_{M+1})$, while the shaded area corresponds to the variance in the prediction.



(a) Prediction point τ_0.

(b) Prediction point τ_1.

(c) Prediction point τ_2.

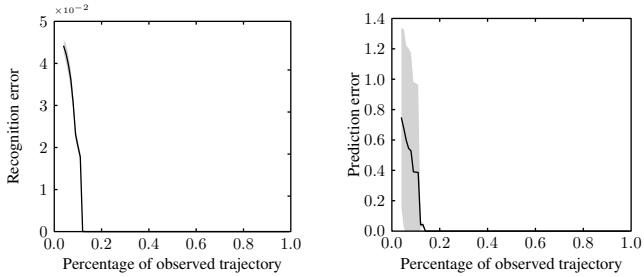
(d) Prediction point τ_3.

(e) Prediction point τ_4.

(f) Final trajectory.

Fig. 4: Illustrative example of the prediction process, as larger portions of the trajectory are observed by the predictor. The red dot corresponds to the mean prediction and the ellipsoid to the corresponding variance.

To further illustrate the prediction process, Fig. 4 showcases the evolution of the prediction as new critical points are observed for a particular trajectory. We can see how the distribution area changes and gets smaller as the observed part of the trajectory increases, until finally arriving at the target point (the mean of this distribution).



(a) Recognition error, given by $|\omega_{\max} - \omega_*|$, with $\omega_{\max}$ is the weight associated with the most likely DMP, and ω_* is the weight for the *correct* DMP.

(b) Prediction accuracy, given by $\|\hat{y}(\tau_{M+1}) - y(\tau_{M+1})\|$, where $\hat{y}(\tau_{M+1})$ is the predicted endpoint and $y(\tau_{M+1})$ is the actual end point.

Fig. 5: The errors in recognition and prediction for the 25 observed trajectories, as a function of the observed trajectory. The solid lines depict the mean error values, while the shaded area depicts the variance in these errors.

C. Uncertainty and anticipatory movement

While observing the trajectory, a key question is: when can the robot take the decision and start its response? As discussed before, the robot will make the decision when the possibility of failure is small, or when there is little variation between the possible predictions considered by the robot. In terms of our approach, this translates into the situations where the distribution of the predicted targets has small variance. By computing the entropy of the recognition weights, we can have an estimate of the uncertainty in our predictions. As discussed in Section II-C, we adopt a threshold ϵ and allow the robot to act in an anticipatory manner if the entropy is below such specified threshold. In our experiments, the threshold value is constant and was selected experimentally [18].

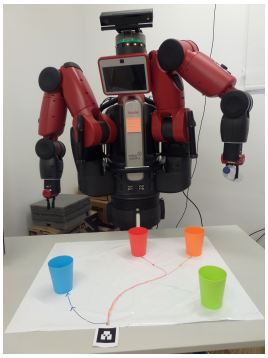
Figure 2b depicts how the entropy in the prediction evolves as we observe larger portions of the trajectory. The point where the robot can initiate its anticipatory action will be when the entropy reaches a small value.

D. Accuracy

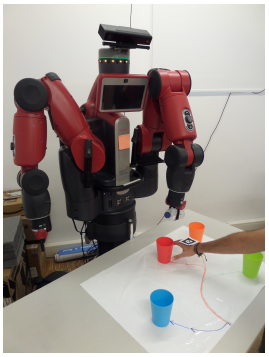
To answer our second question and assess the robustness of our framework, we run a statistical study to check the recognition and prediction and know the accuracy of our approach evolves as larger portions of a trajectory are observed.

The tests in the previous experiments were obtained with a single observed trajectory, while fixing the number of DMPs in the library to 5 DMPs. We now consider a larger library comprising 85 distinct DMPs. We then performed 25 different trajectories, each generated from one DMPs in the library, but modulating it to different targets and execution speeds. We compute the error in recognition and prediction as the different trajectories unfold, and report its mean and variance over the percentage of trajectory observed.

Figure 5 depicts our results. The solid lines represent the mean error, while the shaded areas correspond to the variance in the observed prediction/recognition error. As can be observed from the plot, both the recognition and prediction



(a)



(b)

Fig. 6: Illustration of the human-interaction scenario considered in the paper, where BAXTER moves in anticipation to the user to fill the cup that the user is aiming for.

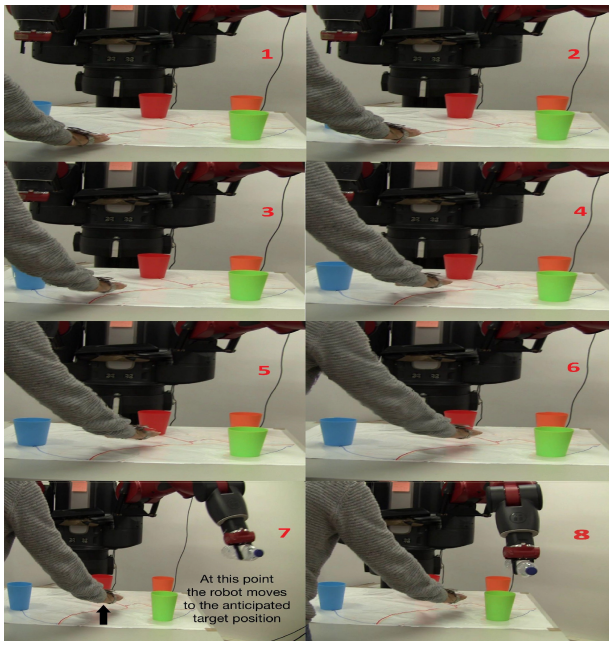


Fig. 7: Successive frames depicting the movement of the user towards the orange cup. When it becomes clear to BAXTER that the target is the orange cup, the robot initiates its own anticipatory movement. At the point indicated by the black arrow, the robot moves to the anticipated target position.

error quickly converge to zero, as does the variance in the prediction. This establishes the ability of our approach to identify and predict the target of an observed trajectory.

E. Implementation in the BAXTER *robot*

We now describe the application of our proposed approach in a human-robot interaction scenario involving a BAXTER robot. In this task, the human user approaches one of the cups in the table (Fig. 6a). Upon realizing which cup the human is reaching, BAXTER performs an anticipatory move towards such cup, "pouring" water in that cup (Fig. 6b). We use the recognition and prediction approaches in Section II.

Figure 7 depicts a sequence of frames that illustrate the our approach at work in the actual robot. The DMPs in the robot library correspond to the motions represented in the table in different colors. While the initial motion of the user could head to any of the red, orange or green cups, the robot remains still, as its uncertainty regarding the target is above the specified threshold. However, as soon as the user's hand moves past the common segment of the three movements, the robot immediately identifies the movement as belonging to the DMP that leads to the orange cup, and initiates the corresponding anticipatory motion, succeeding in "serving" the orange cup before the user actually reaches it.[4]

IV. CONCLUSION

We present in this work a robust and simple method for recognition and prediction of the target of an observed trajectory. Our approach is applicable within the DMP framework, without requiring an exhaustive comparison of all the points in the trajectory, by considering only the critical points. The consideration of critical points alleviates the need for accurate time alignment, and significantly decreases the computational burden associated with both recognition and prediction. Moreover, as we noted in our discussion, often observing intermediate positions in the trajectory provides little added information for recognition or prediction. In contrast, critical points do provide unique data that contributes significantly to disambiguate between the different movements in the library, thus allowing efficient recognition and prediction.

Our work is robust to changes in the observed trajectory when compared to the original DMPs in the library, as it relies on comparing the ratios of change of velocity and angle, dealing efficiently with rotations, modulations and scaling of the trajectory. This is particularly important when dealing with human users, since it is likely that the execution deviates from the exact motion in the robot's library.

Our statistical study illustrates the accuracy of this method in recognition and prediction—our results clearly illustrate the close relation between the performance in terms of prediction and the accuracy in recognition: correct recognition leads to accurate prediction. Finally, our approach is computationally efficient, and can be performed in run-time, a key feature given its relevance in human-robot interaction tasks such as that presented in our experiments.

In this work, we always consider that the movement observed corresponds to a single movement in the library. An interesting direction for future work is to consider situations where the observed work is a composition of movements in the library (e.g., obtained with the framework in [19]).

ACKNOWLEDGEMENTS

This work was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under projects UIDB/50021/2020 and PTDC/CCI-COM/7203/2020. The first author acknowledges the Ph.D. grant from the Global Platform for Syrian Students.

[4]It is worth noting that, in our setup, the computational bottleneck was in the detection and recognition of the human hand, which required the human motion to be executed slowly.

REFERENCES

[1] A. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[2] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in human-humanoid interaction," *Int. J. Robotics Research*, vol. 29, no. 13, pp. 1684–1704, 2010.

[3] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[4] Y. Tanaka, J. Kinugawa, Y. Sugahara, and K. Kosuge, "Motion planning with worker's trajectory prediction for assembly task partner robot," in *Proc. 2012 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2012, pp. 1525–1532.

[5] H. Yoon, J. Soh, Y. Bae, and H. Yang, "Hand gesture recognition using combined features of location, angle and velocity," *Pattern Recognition*, vol. 34, no. 7, pp. 1491–1501, 2001.

[6] M. Black and A. Jepson, "Recognizing temporal trajectories using the condensation algorithm," in *Proc. 3rd IEEE Int. Conf. Automatic Face and Gesture Recognition*, 1998, pp. 16–21.

[7] ——, "A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions," in *Proc. 10th Eur. Conf. Computer Vision*, 1998, pp. 909–924.

[8] M. Kherallah, L. Haddad, A. Alimi, and A. Mitiche, "On-line handwritten digit recognition based on trajectory and velocity modeling," *Pattern Recognition Letters*, vol. 29, no. 5, pp. 580–594, 2008.

[9] G. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, and J. Peters, "A probabilistic framework for semi-autonomous robots based on interaction primitives with phase estimation," in *Proc. 12th Int. Symp. Robotics Research*, 2018, pp. 253–268.

[10] O. Dermy, A. Paraschos, M. Ewerton, J. Peters, F. Charpillet, and S. Ivaldi, "Prediction of intention during interaction with iCub with probabilistic movement primitives," *Frontiers in Robotics and AI*, vol. 4, p. 45, 2017.

[11] G. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, "Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks," *Autonomous Robots*, vol. 41, no. 3, pp. 593–612, 2017.

[12] C. Pérez-D'Arpino and J. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *Proc. 2015 IEEE Int. Conf. Robotics and Automation*, 2015, pp. 6175–6182.

[13] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proc. 2013 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2013, pp. 299–306.

[14] J. Mainprice, R. Hayne, and D. Berenson, "Goal set inverse optimal control and iterative re-planning for predicting human reaching motions in shared workspaces," *IEEE Trans. Robotics*, vol. 32, no. 4, pp. 897–908, 2016.

[15] M. Prada, A. Remazeilles, A. Koene, and S. Endo, "Dynamic movement primitives for human-robot interaction: Comparison with human behavioral observation," in *Proc. 2013 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2013, pp. 1168–1175.

[16] F. Meier, E. Theodorou, and S. Schaal, "Movement segmentation and recognition for imitation learning," in *Proc. 15th Int. Conf. Artificial Intelligence and Statistics*, 2012, pp. 761–769.

[17] H. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *Proc. 2014 IEEE Int. Conf. Robotics and Automation*, 2014, pp. 2831–2837.

[18] B. Akgün, D. Tunaoglu, and E. Sahin, "Action recognition through an action generation mechanism," in *Proc. 10th Int. Conf. Epigenetic Robotics*, 2010.

[19] A. Kordia and F. Melo, "An end-to-end approach for learning and generating complex robot motions from demonstration," in *Proc. 16th IEEE Int. Conf. Control, Automation, Robotics and Vision*, 2020, pp. 1008–1014.